

Deepseek R1

I. Overview:

Nói chung mình skip hết về mấy pha kết quả của nó trên các benchmark, so sánh với các model khác hay là thời gian nó train... vì mấy cái này được nói khá nhiều trên mạng rồi. Mình chỉ note lại những điểm clutch về mặt kỹ thuật khiến nó go pro như vậy. Nhìn chung có vẻ 3 điểm clutch nhất là:

1. Chain of thoughts
2. RL
3. Model distillation: nhằm tạo ra các model nhỏ hơn mà vẫn có thể gần pro như Deepseek-R1.

RL có vẻ như là MVP luôn vì nó nói rất nhiều về RL, nhưng RL cũng dùng data dạng chain of thoughts khá nhiều.

FAQ: làm sao để nó có thể suy luận: do RL + data train đặc thù của nó!

II. Nói qua về kỹ thuật

1. Chain of thoughts

Kiểu là 1 prompt technique để bắt model think out loud (để ý khi dùng deepseek, nó sẽ có 1 đoạn giải thích ở trên trước khi ném ra cái output cho mình), hay nói cách khác nó kiểu như explain step by step.

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both ...

$$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

...

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be ...

We started with the equation:

$$\sqrt{a - \sqrt{a + x}} = x$$

First, let's square both sides:

$$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$$

Next, I could square both sides again, treating the equation: ...

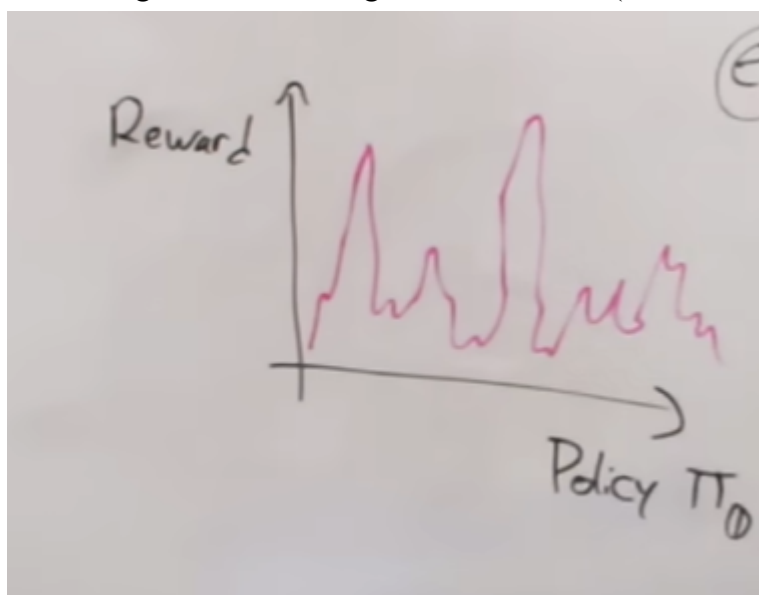
...

Đó và khi sai ở step nào, ta có thể đánh dấu ở chỗ đó xong bắt nó lý giải lại từng bước 1, nhìn chung cách này sẽ cho ra kqua tốt hơn nếu không dùng CoT.

2. RL trong deep seek

RL deepseek dùng theo kiểu cơ chế policy reward

- Policy kiểu how the model behave, reward là điểm thưởng ở thời gian đó. Mục tiêu là tìm policy sao cho max reward.
- Hiểu đơn giản thì kiểu policy là cho ngoại cảnh, model làm gì, và ở mỗi tình huống đó điểm thưởng nó là bao nhiêu (nhìn hình bên dưới).



2.1. RL Algorithm: Group Relative Policy Optimization

Chỗ này 1 tỷ toán nhưng nch nó là cái công thức để optimize policy với reward thôi

Group Relative Policy Optimization In order to save the training costs of RL, we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which foregoes the critic model that is typically the same size as the policy model, and estimates the baseline from group scores instead. Specifically, for each question q , GRPO samples a group of outputs $\{o_1, o_2, \dots, o_G\}$ from the old policy $\pi_{\theta_{old}}$ and then optimizes the policy model π_{θ} by maximizing the following objective:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G} \sum_{i=1}^G \left(\min \left(\frac{\pi_{\theta}(o_i|q)}{\pi_{\theta_{old}}(o_i|q)} A_i, \text{clip} \left(\frac{\pi_{\theta}(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1 - \epsilon, 1 + \epsilon \right) A_i \right) - \beta \mathbb{D}_{KL}(\pi_{\theta} || \pi_{ref}) \right), \quad (1)$$

$$\mathbb{D}_{KL}(\pi_{\theta} || \pi_{ref}) = \frac{\pi_{ref}(o_i|q)}{\pi_{\theta}(o_i|q)} - \log \frac{\pi_{ref}(o_i|q)}{\pi_{\theta}(o_i|q)} - 1, \quad (2)$$

where ϵ and β are hyper-parameters, and A_i is the advantage, computed using a group of rewards $\{r_1, r_2, \dots, r_G\}$ corresponding to the outputs within each group:

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \dots, r_G\})}{\text{std}(\{r_1, r_2, \dots, r_G\})}. \quad (3)$$

Cái này khác cái bình thường người ta đang dùng vì nó tính reward theo nhóm.

2.2: Dùng RL lên base model bằng cái thuật toán dùng ở 2.1 kia, tạo ra Deepseek-R1-Zero

Kỹ thuật này được sử dụng để tạo ra cái model Deepseek-R1-Zero, nhưng bọn này không dùng 1 tý Supervised data nào vì cno bảo mất time thu thập, thay vào đó dựa hoàn toàn vào khả năng của model (without any supervised data, focusing on their self-evolution through a pure reinforcement learning process). Xem trang 5 paper deepseek-r1.

Dưới đây là cái data cno dùng ở đoạn này:

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within `<think>` `</think>` and `<answer>` `</answer>` tags, respectively, i.e., `<think>` reasoning process here `</think>` `<answer>` answer here `</answer>`. User: **prompt**. Assistant:

Table 1 | Template for DeepSeek-R1-Zero. **prompt** will be replaced with the specific reasoning question during training.

Về điểm thưởng reward, cno chơi **reward theo rule base** chứ không neural base: kiểu nói chung nó sẽ định nghĩa 1 cái hàm reward (theo mình hiểu là thế), dựa trên 2 yếu tố:

- **Accuracy reward**: điểm thưởng cho độ chính xác, thường cno dùng mấy bài toán chỉ có 1 đáp án duy nhất hoặc kiểm tra mã leetcode bằng trình biên dịch (ồ lỗi hồng ở đây là có thể code không tối ưu nhé vì nó chỉ cần đúng thôi).
- **Format reward**: đảm bảo suy luận nằm trong cặp token `<think>`, trả lời trong `<answer>`.

Nói chung có vẻ Deepseek-R1-Zero, ý của từ Zero là RL thăng lên base model bằng cái data theo template bên trên kia luôn, data thì không có hướng dẫn cụ thể nó phải gen gì trong các token `<think>` và `<answer>` (hoàn toàn không supervised), nên cái đoạn train này cno gọi là **Self-evolution Process of Deepseek-R1-Zero** (tự tiến hóa).

Một điều thú vị là step train càng cao lên thì model tự phát ra mấy skill khá hay ho kiểu thức tỉnh huyết mạch (hoàn toàn không có tác động từ bên ngoài, không có trong data vì như trên data hoàn toàn cho nó gen tự do trong token `<think>`):

- **suy nghĩ phản ánh (reflection)** — nơi mô hình xem xét lại và đánh giá các bước trước đó — và
- **khám phá các cách tiếp cận khác nhau** để giải quyết vấn đề xuất hiện một cách tự phát.

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both ...

$$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

...

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be ...

We started with the equation:

$$\sqrt{a - \sqrt{a + x}} = x$$

First, let's square both sides:

$$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$$

Next, I could square both sides again, treating the equation: ...

...

Kiểu này, ngta đánh dấu đoạn này là **“aha moment”** ý chỉ tự dung model thực hiện 1 pha double check cho bài làm của mình một cách rất ngẫu nhiên. Đây có thể là bước ngoặt cho 1 thứ to hơn trong tương lai.

2.3: Điểm yếu của Deepseek-R1-Zero

Theo báo nói thì điểm yếu là cái nó gen ra khó đọc hiểu và trộn lẫn nhiều ngôn ngữ, nên người ta nghĩ ra thêm cách vẫn RL nhưng được khởi động với dữ liệu có giám sát ban đầu (cold-start data) được thiết kế thân thiện với con người (human-friendly).

Cái dữ liệu này nói thẳng ra chính là các CoT nhưng thân thiện với con người, và đó là cách ngta tạo ra Deepseek-R1.

2.4: DeepSeek-R1: Reinforcement Learning with Cold Start

Nói chung nó chia phần này ra làm hẳn 4 giai đoạn, mình sẽ giải thích ngắn gọn về từng giai đoạn:

Model base dùng ở đây là Deepseek-V3-Base

Phase 1: Cold Start

- RL khởi động với data CoT dài và chất lượng (khá tốn công gán nhãn).
- Cách collect:
 - few-shot prompting dài
 - prompt thêm khuyến khích model tự phản ánh và tự kiểm tra
 - lấy kết quả của R1-Zero nhưng rephrase cho dễ đọc.
 - post processing bằng người.

Nhờ cold-start, đầu ra nó trở thành:

- `|special_token|<reasoning_process>|special_token|<summary>`
 - `<reasoning_process>`: Là chuỗi suy luận từng bước dựa trên phương pháp Chain-of-Thought (CoT).
 - `<summary>`: Phần tóm tắt, giúp làm rõ kết quả cuối cùng của quá trình suy luận, giúp người đọc hiểu nhanh hơn.

Phase 2: Reasoning-oriented Reinforcement Learning

Phần này làm như trên đã trình bày với cái Deepseek-R1-Zero, nhưng nó cho thêm 1 cái reward gọi là *phần thưởng cho sự nhất quán về ngôn ngữ*.

Phase 3: Rejection Sampling and Supervised Fine-Tuning

Đoạn này cần tạo 1 bộ data SFT sau đó SFT checkpoint bên trên, nhìn chung cách tạo là rejection sampling:

- Dùng checkpoint bên trên sample ra 1 bộ data.
- Loại bỏ mấy mẫu data bị lỗi, điển hình như bị trộn 1 đồng ngôn ngữ trong câu trả lời.

Điểm mới nữa là nó thêm vào các data có thể đánh giá được bằng metrics chứ không phải chỉ độc rule-base reward như ở trên. Nhưng nói chung ngta cũng mô tả cách tạo thành 2 bộ data Reasoning và non-reasoning, reasoning thì tạo như mô tả bên trên, còn non reasoning kiểu mấy task như writing, factual QA, self-cognition, translation, nó reuse data của bọn deepseek V3.

Trong 1 số trường hợp, kể cả task non-reasoning, nó prompt bắt model viết 1 đoạn CoT ngắn ngắn ra trước khi trả lời để lấy làm data.

Phase 4: Reinforcement Learning for all Scenarios

Đoạn này nó cố gắng làm mô hình gen ra kết quả hợp với sở thích con người, bằng cách thêm nhiều điểm thưởng theo các cách khác nhau, vdu như đoạn nào được người dùng thích hơn, hay đoạn tóm tắt nào thực sự hữu ích (cái này trông cũng gán nhãn bằng người).

3. Distillation

Nói chung cái distil này là 1 chủ đề khá dài, có lẽ sẽ viết riêng ở 1 chỗ khác sau!